

IN THE CLAIMS:

Please Amend the claims to read as follows:

Please cancel claims 7-14, 16-19, and 33 without prejudice.

1 1. (Previously Presented) A method for converting a file access data structure from a
2 first endianness to a second endianness by a processor, the method comprising the steps
3 of:
4 identifying, from a descriptor look up table, a series of actions to perform on ele-
5 ments of the file access data structure, where the series of actions include at least one of
6 converting, copying, or linking; and
7 performing the identified series of actions on the elements of the file access data
8 structure to convert the file data structure from the first endianness to the second endian-
9 ness.

1 2. (Previously Presented) A method of converting elements of a file access data structure
2 from a first endianness to a second endianness by a processor, the method comprising the
3 steps of:
4 determining if the file access data structure is a critical path data structure;

5 converting, in response to the file access data structure being a critical path data
6 structure, the elements from the first endianness to the second endianness using a set of
7 specific code functions;

8 converting, in response to the file access data structure not being a critical path
9 data structure, a header of the file access data structure from the first endianness to the
10 second endianness using a second set of specific code functions; and

11 calling a byte swapping engine to convert selected elements of the file access data
12 structure from the first byte order to the second byte order.

1 3. (Original) The method of claim 2 wherein the file access data structure further com-
2 prises a direct access file access data structure.

1 4. (Previously Presented) A file system for converting elements of a file access data
2 structure from a first endianness to a second endianness, the system comprising:

3 an input buffer, the input buffer storing the file access data structure with the first
4 endianness to be converted;

5 a byte swapping engine, the byte swapping engine operative interconnected with a
6 descriptor table, with the descriptor table listing a series of actions to perform when con-
7 verting the file data structure from the first endianness to the second endianness, where
8 the series of actions include at least one of converting, copying, or linking; and

9 an output buffer, the byte swapping engine placing the file access data structure
10 with the second endianness in the output buffer after conversion.

1 5. (Original) The system of claim 4 wherein the descriptor table further comprises a set
2 of entries describing various file access data structures, each entry further comprising a
3 size field and an operation field.

1 6. (Original) The system of claim 4 wherein the file access data structure further com-
2 prises a direct access file access data structure.

7-14 (Cancelled)

1 15. (Previously Presented) A computer-readable medium, including program instructions
2 executing on a computer, for converting elements of a file access data structure from a
3 first endianness to a second endianness, the method comprising the steps of:
4 determining if the file access data structure is a critical path data structure;
5 converting, in response to the file access data structure being a critical path data
6 structure, the elements from the first endianness to the second endianness using a set of
7 specific code functions;
8 converting, in response to the file access data structure not being a critical path
9 data structure, a header of the file access data structure from the first endianness to the
10 second endianness using a second set of specific code functions; and
11 calling a byte swapping engine to convert selected elements of the file access data
12 structure from the first byte order to the second byte order.

16-19 (Cancelled)

1 20. (Previously Presented) A method for converting a data structure by a processor,
2 comprising:
3 calling a byte-swapping engine;
4 providing a file access data structure as input to the byte-swapping engine;
5 providing a descriptor look up table to the byte-swapping engine;
6 identifying, from the descriptor look up table, a series of actions to perform on
7 elements of the file access data structure in order to swap bytes of the file access data
8 structure from a first endianness to a second endianness, where the series of actions in-
9 clude at least one of converting, copying, or linking; and
10 performing the identified series of actions on the elements of the file access data
11 structure to convert the file access data structure.

1 21. (Previously Presented) The method as in claim 20, further comprising:
2 using as the file access data structure a file having Direct Access File System
3 (DAFS) protocol.

1 22. (Previously Presented) The method as in claim 20, further comprising:
2 determining if the file access data structure is a critical path data structure, where
3 the critical path data structure includes commonly utilized data structures, and if the file

4 access data structure is a critical path data structure, perform byte swap operations using
5 specific code functions.

1 23. (Previously Presented) The method as in claim 20, further comprising:

2 determining if the file access data structure is a critical path data structure, where
3 the critical path data structure includes commonly utilized data structures, and if the file
4 access data structure is not a critical path data structure, perform byte swap operations on
5 a data structure header.

1 24. (Previously Presented) The method as in claim 20, further comprising:

2 swapping bytes of the data structure as needed, in response to swapping bytes of
3 the file access data structure.

1 25. (Previously Presented) The method as in claim 20, further comprising:

2 determining if an element entry of the descriptor look up table is nested;
3 branching to the nested entry;
4 identifying, from the descriptor look up table, a nested series of actions to perform
5 on elements of the nested entry in order to swap bytes of the entry from a first endianness
6 to a second endianness, where the nested series of actions includes linking and convert-
7 ing.

1 26. (Previously Presented) A computer to convert a data structure by a processor, com-
2 prising:

3 means for calling a byte-swapping engine;

4 means for providing a file access data structure as input to the byte-swapping en-
5 gine;

6 means for providing a descriptor look up table to the byte-swapping engine;

7 means for identifying, from the descriptor look up table, a series of actions to per-
8 form on elements of the file access data structure in order to swap bytes of the file access
9 data structure from a first endianness to a second endianness, where the series of actions
10 include at least one of converting, copying, or linking; and

11 means for performing the identified series of actions on the elements of the file
12 access data structure to convert the file access data structure.

1 27. (Previously Presented) The computer as in claim 26, further comprising:

2 means for using as the file access data structure a file having Direct Access File
3 System (DAFS) protocol.

1 28. (Previously Presented) The computer as in claim 26, further comprising:

2 means for determining if the file access data structure is a critical path data struc-
3 ture, where the critical path data structure includes commonly utilized data structures, and
4 if the file access data structure is a critical path data structure, perform byte swap opera-
5 tions using specific code functions.

1 29. (Previously Presented) The computer as in claim 26, further comprising:
2 means for determining if the file access data structure is a critical path data struc-
3 ture, where the critical path data structure includes commonly utilized data structures, and
4 if the file access data structure is not a critical path data structure, perform byte swap op-
5 erations on a data structure header.

1 30. (Previously Presented) The computer as in claim 26, further comprising:
2 means for swapping bytes of the data structure as needed, in response to swapping
3 bytes of the file access data structure.

1 31. (Previously Presented) The computer as in claim 26, further comprising:
2 means for determining if an element entry of the descriptor look up table is
3 nested;
4 means for branching to the nested entry;
5 means for identifying, from the descriptor look up table, a nested series of actions
6 to perform on elements of the nested entry in order to swap bytes of the entry from a first
7 endianness to a second endianness, where the nested series of actions includes converting
8 and linking.

1 32. (Previously Presented) A computer readable media, comprising:

2 said computer readable media containing instructions for execution on a processor
3 for the practice of a method for converting a data structure by a processor, the method
4 having the steps of,
5 calling a byte-swapping engine;
6 providing a file access data structure as input to the byte-swapping engine;
7 providing a descriptor look up table to the byte-swapping engine;
8 identifying, from the descriptor look up table, a series of actions to perform on
9 elements of the file access data structure in order to swap bytes of the file access data
10 structure from a first endianness to a second endianness, where the series of actions in-
11 clude at least one of converting, copying, or linking; and
12 performing the identified series of actions on the elements of the file access data
13 structure to convert the file access data structure.

1 33. (Cancelled)

1 34. (Previously Presented) A method of converting elements of a file access data struc-
2 ture from a first endianness to a second endianness by a processor, comprising:

3 determining if the file access data structure is a critical path data structure; and

4 converting the elements from the first endianness to the second endianness using a
5 set of specific code functions if the file access data structure is a critical path data struc-
6 ture.

1 35. (Previously Presented) The method of claim 34, further comprising:
2 converting a header of the file access data structure from the first endianness to
3 the second endianness using a second set of specific code functions if the file access data
4 structure is not a critical path data structure.

1 36. (Previously Presented) The method of claim 34, further comprising:
2 calling a byte swapping engine to convert selected elements of the file access data
3 structure from the first byte order to the second byte order.

1 37. (Previously Presented) A method for converting a first data structure from a to a sec-
2 ond data structure by a processor, the method comprising the steps of:
3 using a descriptor lookup table to provide actions to be performed on each ele-
4 ment of the first data structure; and
5 stepping through the descriptor table and processing each element of the first data
6 structure according to the element's size and action to convert the first data structure into
7 the second data structure.

1 38. (Previously Presented) The method of claim 37, further comprising:
2 using a byte as the data structure.

1 39. (Previously Presented) The method of claim 2, wherein the critical data path structure
2 includes commonly used data structures.

1 40. (Previously Presented) The method of claim 2, wherein the critical data path structure
2 is a direct access file system (DAFS) header data structure.

1 41. (Previously Presented) The method of claim 2, wherein the specific code functions
2 are designed to rapidly convert any elements of the data structure to the second endian-
3 ness without using a byte swapping engine.

1 42. (Previously Presented) The computer-readable medium of claim 15, wherein the criti-
2 cal data path structure includes commonly used data structures.

1 43. (Previously Presented) The computer-readable medium of claim 15, wherein the criti-
2 cal data path structure is a direct access file system (DAFS) header data structure.

1 44. (Previously Presented) The computer-readable medium of claim 15, wherein the spe-
2 cific code functions are designed to rapidly convert any elements of the data structure to
3 the second endianness without using a byte swapping engine.

1 45. (Previously Presented) The method of claim 34, wherein the critical data path struc-
2 ture includes commonly used data structures.

1 46. (Previously Presented) The method of claim 34, wherein the critical data path struc-
2 ture is a direct access file system (DAFS) header data structure.

1 47. (Previously Presented) The method of claim 34, wherein the specific code functions
2 are designed to rapidly convert any elements of the data structure to the second endian-
3 ness without using a byte swapping engine.